Laid-open Publication No. JP 2-99746

Date of Publication : April 11, 1990

1. Title of the Invention

Engine control device

2. Claim(s)

1. An engine control device having a random access memory (RAM) for storing data, a read only memory (ROM) for storing a program, and a central processing unit (CPU) for processing data by the program stored in this ROM, the engine control device being designed to compute at least one data indicating an operational status of an engine by the program to thereby create data for controlling the engine,

wherein

the engine control device comprises

a program-writable ROM which becomes available for writing a program under a specific operating condition; and

a ROM in which a program served for externally writing data in the program-writable ROM is stored, and

wherein

arithmetic processings for controlling the engine are conducted by the externally-written program in the program-writable ROM.

2. The engine control device according to claim 1, wherein in the program-writable ROM which becomes available for writing a program under a specific operating condition, writing and reading is switched by an instruction of the CPU.

3. The engine control device according to claim 1, wherein a second vector address is provided in a predetermined address of the

BEST AVAILABLE COPY

program-writable ROM which becomes available for writing a program under a specific operating condition.

3. Detailed Description of the Invention

[Industrial Field of Utilization]

The present invention relates to a control device utilizing a microcomputer that is utilized for an internal combustion engine such as a gasoline engine, more specifically relates to the control device for an automobile gasoline engine.

[Prior Art]

An engine control device utilizing a microcomputer is desirably able to select each of controlling data and programs in accordance with aging changes and properties of the engine to which the engine control device is assembled.

A conventional engine control device is proposed by the invention of Japanese Patent Application Laid Open No. S62-33356. In the known method, a program is externally read and stored in RAM (Random Access Memory) for inspection of the control device itself and engine matching, and thereby the engine control is conducted.

[Problems to be solved by the Invention]

Various parts on a vehicle are exposed to severe circumstances such as temperature and vibration. To cope with this, an engine control device needs to have improved vibration-resistant properties. Because of this, a socket cannot be utilized even for attachment of the IC. This situation is the same as that of ROM (Read Only Memory). Before assembling the control device, a program is stored in the ROM by a special machine, and the ROM is directly soldered on a circuit board.

On the other hand, the engine control device might be caught up in a

trouble where the contents of the ROM must be changed because of the mismatching of programs and control data which leads to an on market trouble. The ROM already attached must be removed to be replaced to a new one by soldering. Accordingly, the conventional technology suffers from problems of increased man-hours and deterioration of reliability.

In addition, the foregoing invention of Japanese Patent Application Laid Open No. S62-33356 discloses a method for writing a program in RAM for execution of it. Thus, this invention is suitable for inspection and matching purposes, but not suitable to the unexpected change of the contents of the ROM caused by the on market trouble.

The object of the present invention is to provide an engine control device capable of directly writing a new program in the ROM in a soldered condition on the circuit board.

[Means for Solving the Problems]

The foregoing object is achieved by utilizing serial communication with an external device and writing a program externally read in the ROM.

[Function]

The ROM built in the CPU (Central Processing Unit) serves to store only an executable program to implement a program externally read via serial communication that is stored in RAM. The executable program serves to load the program that should be stored in ROM in the RAM. If the program stored in the RAM is executed, the CPU send a signal to the ROM to allow the ROM to be a writable mode. The data (programs) which have been input via the serial communication are sequentially written in the ROM. This permits to rewrite the programs without the need of exchanging the ROM.

[Preferred Embodiments]

With reference to drawings, the embodiments explain the engine control device in more detail.

FIG. 1 is one embodiment of the present invention. In this figure, a referential numeral 1 designates a CPU for performing various operations and processes. The CPU 1 has a built-in mask ROM 6 with erase- and rewrite- protection. Referential numeral 2 designates a ROM served for storing a program and data used for reading, processing, and operating by the CPU 1. Referential numeral 3 designates RAM served for reading or writing the data computed by the CPU 1. Referential numeral 4 designates I/O for inputting input signals from various sensors and outputting various control signals. Referential numeral 5 designates a decoder for outputting a signal for accessing each device. Each of the CPU 1, ROM 2, RAM 3, and I/O 4 is connected to an address bus 7 and a data bus 8. In the foregoing configuration, input to the I/O 4 are a rotation signal 4a for an engine revolution speed and an intaking-air volume signal 4b as an input signal from a sensor for detecting an intake-air volume by the engine. The CPU 1 computes the injection quantity to the engine using these input signals. A fuel-injection control signal 4c is output from the I/O 4 based on the result of computation. The operations as basic control in controlling the engine is as explained above. In addition, control items can be added by increasing input and output.

Although a program necessary for the foregoing control is stored in ROM 2, the ROM 2 of this embodiment has the properties of the erase-protection even if the power supply of a control device is turned OFF and the rewrite-protection during operation. This erase- and rewrite-protection of the ROM 2 is realized in such a way that a PGM signal 2a of the ROM 2 is set as a power supply voltage Vcc (normally 5 V) and a Vpp

signal 2b is set as Vcc. Incidentally, radiation of an ultraviolet ray may erase the contents of the memory of ROM 2. Further ROM 2 can be written if a PGM signal 2a is set as Low and a voltage of 12.5 or so is applied to a Vpp signal.

Referential numeral 5a designates a CS signal as a decode signal for a decoder 5 to access the ROM 2. This CS signal 5a turns to be Low only if the ROM 2 is accessed regardless of reading-out or writing-in. In other cases, the CS signal 5a outputs High.

The referential numeral 1a designates a mode switch signal output from the CPU 1 to the ROM 2. Namely, this mode switch signal 1a is normally Low. Both of an NPN transistor 10 and a PNP transistor 11 are turning to be OFF, and the voltage supplied to the Vpp signal 2b becomes Vcc by a pull up resistance 12. Further, the PGM signal 2a as an output signal from a NAND circuit 9 is High because the mode switch signal 1a is Low.

On the other hand, in the case of writing a program in the ROM 2, if the mode switch signal 1a is turned to be High, both of the NPN transistor 10 and the PNP transistor 11 are turned to be ON and thereby a voltage of Vb (≈ 12.5 V) is applied to the Vpp signal 2b. Further, the output from the NAND circuit 9 is the same as that of the CS signal 5a. Namely, although the referential numeral 5a is input to the NAND circuit 9 via a NOT circuit 13, since the mode switch signal 1a as another input is High, the output signal from the NOT circuit 13 is further reversed to be output. Accordingly, in the event that the CS signal 5a is Low, namely the ROM 2 is accessed, the PGM signal 2a turns to be Low and the ROM 2 functions as a writable mode. In the writable mode, data output to the data bus 8 is written at an address designated by the address bus 7.

Three input / output signal lines for serial communication with external are provided in the CPU 1. Referential numeral 1b designates a SCI clock signal as a sending / receiving basic clock. Referential numeral 1c designates a SCI receiving signal served for the CPU 1 to receive the data sent externally. Referential numeral 1d designates a SCI sending signal for sending externally. The Japanese Patent Application Laid Open No. S62-33356 describes each signal timing; the first bit is a start bit and the last bit is a stop bit.

FIG. 2 is an address allocation table showing an example of allocating addresses for the embodiment. Referential numeral 20 designates addresses for various registers for switching the internal functions of the CPU 1. Referential numeral 21 designates addresses ($1000 - $17FF) in the RAM 3. Referential numeral 22 designates addresses ($8000 - $BFFF) in the ROM 2. Referential numeral 23 designates addresses ($C000 - $FFFF) of the ROM 6 built in the CPU 1. Referential numeral 23a designates vector addresses for designating a jump address at the time of any interruption by a reset, SCI, etc. For example, after the CPU 1 is reset and the reset condition is released, the jump address is read from such vector addresses $FFFE and $FFFF to be read and thereby a designated routine is executed. Also in the case of the interruption, the operation is carried out by the same sequence. On the other hand, a referential numeric 22a designates a second vector address for designating a jump address for executing programs written in the ROM 2. Namely, if some interruption is occurred and the interrupt service routine is allocated at addresses 22 of the ROM 2, the first address of the interrupt service routine must be inserted into the vector address 23a in the mask ROM 6. In this case, even if the program is changed, the first address

cannot be changed. This means that versatility is sufficient. To cope with this, the address of the second vector address 22a of the ROM 2 is inserted in the vector address 23a to thereby jump into a predetermined routine. FIG. 3 is a flowchart showing an embodiment in the case where the second vector address is used. For example, if NMI as the top priority interruption is occurred at S1, the jump address is read from $FFFC and $FFFD in the vector address 23a at S2 to thereby jump into the second vector address as that address. Note that S1 and S2 depend on the sequence of the hardware and thus cannot be changed by software. S3 designates the contents of processing at the second vector address and an instruction for jumping into the predetermined process routine. S4 designates the execution of the process of NMI routine after jumping into the top address of the NMI routine. S5 designates the continuation of the process before interruption by releasing from interruption after the process of S4.

With this configuration, the second vector address 23a is rewritable, and thus versatility can be kept.

Next, the process of writing to the ROM 2 is explained.

FIG. 4 is a flowchart of the process of a reset routine. An initial setup of a SCSI board is performed at S11 so as to enable SCI to be available. Next, data R1 located at the predetermined address in the RAM 3 are reset at S12. At S13, the process is jumped into the second vector address. The processes at S11 - S13 are executed by the program of the mask ROM 6. At S14, the condition of waiting interruption is maintained.

FIG. 5 is a flowchart of a process in the case of receiving interruption of the serial communication. In particular, this is the operation of writing the program in RAM 3. If SCI interruption is caused, it is checked at S21 to confirm whether or not Prog flag as the predetermined bit of the content R2

at some address of the ROM 2 is 1. This flag is utilized for judging whether or not the mode of writing the current program in the RAM 3 is established.

When Prog flag is zero, receiving data are read at S22. Then, it is judged whether or not the flag ROM of the R1 is 1 at S23. The flag ROM is a flag for indicating that the program is being written in the ROM 2. When the flag ROM is 1, it is deemed that the program for writing to the ROM 2 has already been written in the RAM 3. Then, this program is executed at S29. When the flag ROM is zero at S23, it is deemed that SCSI interruption is a first interruption. This first receiving data is allowed to have the meaning as data which serves to indicates a program writing mode for a RAM 3 or a program writing mode for ROM 2. At S24, it is checked to confirm whether or not this first receiving data designate a ROM writing mode. If in the ROM writing mode, the flag ROM is set to 1 at S28, and then the process of interruption is terminated once. Otherwise, it is checked to confirm whether or not the condition is the program writing mode for RAM 3 at S25. If not, since the condition is another mode, the predetermined process is executed by jumping into the second vector address. However, only this case has a premise that the predetermined program is already installed in the ROM 2. If it is judged that the condition is the program writing mode at S25, the Prog flag is set to 1 at S27, and then the process of interruption is terminated.

If it is confirmed that the Prog flag is 1 at S21, since the current condition is the program writing mode for the RAM 3, the succeeding processes indicate the processes of storing the receiving data in the RAM 3. The condition of a Start flag is checked at S30. The Start flag is a flag of determining whether or not the receiving data is the first receiving data after the program writing mode for the RAM 3 starts. When the Start flag is zero,

the first address for storing the RAM 3 is designated at S31, and then the Start flag is set to 1 at S32. Note that in the program writing mode, the first data designates the number of data which are sequentially sent. After the first data is stored in the predetermined address, data are decremented by each time data are sent. At S33, the number of data sequentially sent is written in a first address, only 1 is decremented, and the content is checked at S34. If the content of the first address is zero, it is deemed that a sending process by the program is terminated, and then the Prog flag and Start flag is set to 0 at S35. An address to which data sent are stored is incremented at S36, the receiving data are read at S37, and then data are stored at the foregoing address at S38.

In this way, the foregoing processes are executed, and thus the program can be written in the RAM 3. At this stage, it is assumed that the program as shown in FIG. 5 is stored in the mask ROM 6 of the CPU 1.

FIG. 6 is a flowchart for a program of writing to the ROM 2, which is written in the RAM 3 by the foregoing operation. In the same way as in the case that the program is written in the RAM 3, the first sending data designates the number of data which will be sent in the future. At S40, the condition of the flag S for judging whether or not the data are the first data as the ROM writing data is checked. In the case of zero, the receiving data are stored at the address of R2 in the RAM 3 at S41. The first address of the ROM is designated at S42. At S43, The process of interruption is terminated by setting 1 to the flag S. If it is confirmed that the data are the second or succeeding data at S40, the content of R2 is decremented at S44 and it is checked whether or not R2 is set to 0 at S45. In the case that R2 is zero, 0 is set to the flag ROM and flag S at S46 in order to indicate that the receiving of the data for writing to the ROM is terminated. The address

at which data are stored in the ROM 2 is incremented at S47, and the mode switch signal 1a is turned to be High at S48. Accordingly, as mentioned above, the ROM 2 turns to be the writable mode. The receiving data are written in the address at which the receiving data are stored at S49. At S50, the writing is repeated for a predetermined time. After the writing is finished, the mode of the ROM 2 is set to a normal mode by turning the mode switch signal 1a to be Low at S51. At S52, in order to confirm that the data have been written correctly, the data are checked to confirm that the data are equal to the receiving data by reading the data at the address at which the data have been stored. When the data are not equal, the writing operation is executed once more by returning to S48. When the data are confirmed that the data are equal at S53, the fact that the operation of writing the receiving data is finished is sent to an external device at S54, and then the process is terminated. The external device sends the next writing data, if the data indicating which the data has been written completely in the ROM are sent.

Therefore, according to the present embodiment, the program can be written in the ROM 2 by utilizing the serial communication, and the operation can be handled easily and successfully even if the program and data are not mismatched at a marketplace.

[Effects of the Invention]

According to the effect of present invention, in changing the contents of the ROM as the countermeasure for on market troubles of the engine control, the ROM can be directly rewritten while the ROM is in a soldered condition with the circuit board. Thus, the man hours for changing the contents of the ROM can be reduced dramatically. Further, since the operation of soldering again is not needed, the degradation of reliability due

to a soldering operation can be attained.

4. Brief Explanation of the Drawings

FIG. 1 is one embodiment of the present invention.

FIG. 2 is an address allocation table.

FIG. 3 is a flowchart of an embodiment of using the vector address of FIG. 2.

FIG. 4 is a flowchart of the process of a reset routine.

FIG. 5 is a flowchart of a process of writing a program in RAM 3 by utilizing serial communication.

FIG. 6 is a flowchart of the operation of writing the program in ROM 2.

1 ······ CPU,

2 ······ ROM,

3 ······ RAM,

4 ······ I/O,

5 ······ Decoder,

6 ······ Mask ROM,

7 ······ Address bus,

8 ······ Data bus,

9 ······ NAND circuit,

10 ······ NPN transistor,

11 ······ PNP transistor,

12 ······ Pull up resistance,

13 ······ NOT circuit,

1a ······ Mode switch signal,

2a ······ PGM signal,

2b ······ Vpp signal,

5a …… CS signal,

1b ……SCI clock signal,

1c …… SCI receiving signal,

1d …… SCI sending signal,

20 …… Addresses of various registers,

21 …… Address of RAM 3,

22 …… Address of ROM 2,

23 …… Address of mask ROM 6,

22a …… Second vector address,

23a …… Vector address,

4a …… Rotation signal,

4b …… Intake-air volume signal,

4c …… Fuel-injection control signal.

FIG. 1

4a: ROTATION SIGNAL

4b: INTAKE-AIR VOLUME SIGNAL

4c: FUEL-INJECTION CONTROL SIGNAL 4C

1a: MODE SWITCH SIGNAL

1b: SCI CLOCK SIGNAL

1c: SCI RECEIVING SIGNAL

1d: SCI RECEIVING SIGNAL

2a: PGM SIGNAL

2b: Vpp SIGNAL

5a: CS SIGNAL

5: DECODER

6: MASK ROM

FIG. 3

S1: NMI INTERRUPTION OCCURRENCE

S2: READ VECTOR ADDRESSES $FFFC,D

S3: JMP NMI

S4: NMI ROUTINE

FIG. 4

PROGRAM OF MASK ROM 6

S11: SCI PORT SETUP

S12: PREDETERMINED RAM (R1) RESET

S13: JUMP TO THE SECOND VECTOR

ADDRESS OF ROM 2

S14: INTERRUPTION WAIT

FIG. 5

S22: RECEIVING DATA READ

S24: ROM WRITING MODE

S25: PROGRAM WRITING MODE

S26: JUMP TO SECOND VECTOR ADDRESS

S29: EXECUTE PROGRAM WRITTEN IN RAM

S31: FIRST ADDRESS DESIGNATION

S33: DECREMENT 1 FROM THE CONTENT AT THE FIRST ADDRESS

S34: CONTENT AT THE FIRST ADDRESS = 0 ?

S36: INCREMENT STORE ADDRESS

S37: READ RECEIVING DATA

S38: STORE DATA IN STORE ADDRESS

FIG. 6

S41: STORE RECEIVING DATA IN R2

S42: DESIGNATE ROM FIRST ADDRESS

S44: DECREMENT 1 FROM CONTENT OF R2

S47: INCREMENT ROM 2 STORE ADDRESS

S48: OUTPUT MODE SWITCH SIGNAL

S49: WRITE DATA AT ROM 2 STORE ADDRESS

S50: PREDETERMINED TIME ?

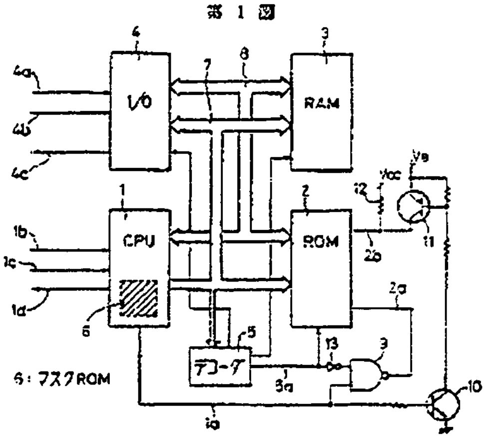
S51: OUTPUT MODE SWITCH SIGNAL LOW

S52: READ ROM 2 STORE ADDRESS

S53: IS EQUAL TO RECEIVING DATA ?

S54: SEND WRITE COMPLETION TO EXTERNAL DEVICE

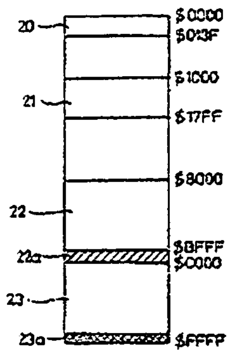
第 1 図



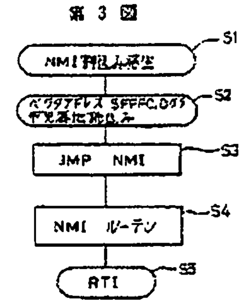
4a : 回転信号
4b : 吸入空気量信号
4c : 燃料噴射制御信号
1a : モード切換信号
1b : SCIクロック信号
1c : SCI受信信号
1d : SCI送信信号

2a : PGM信号
2b : Vpp 信号
5a : CS 信号

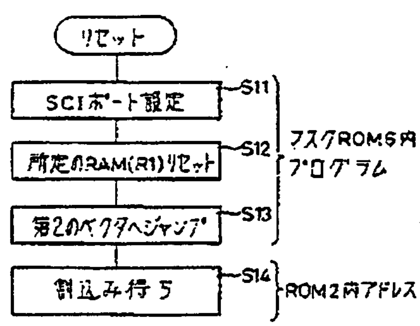
第 2 図



第 3 図

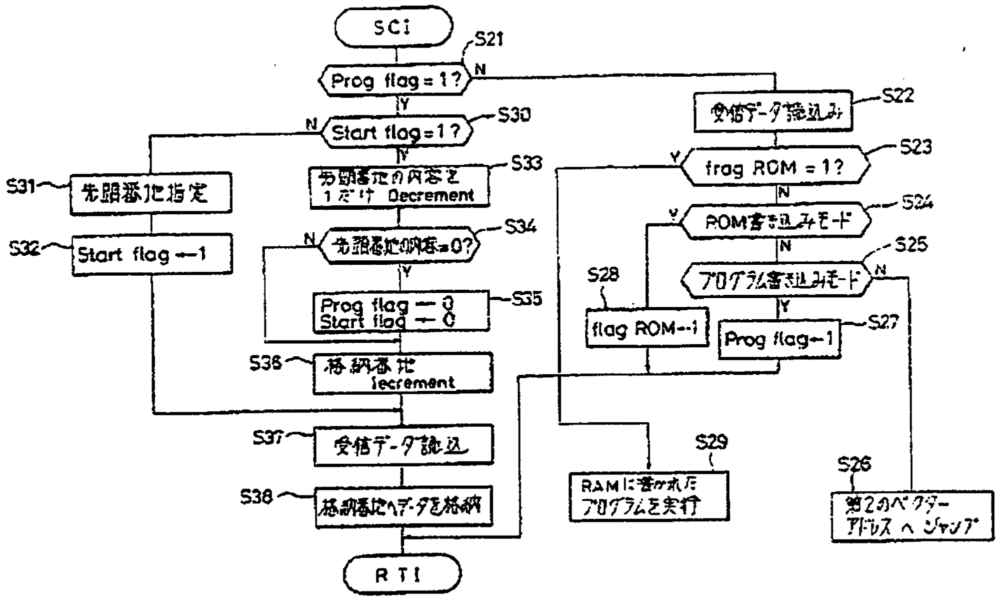


第 4 図

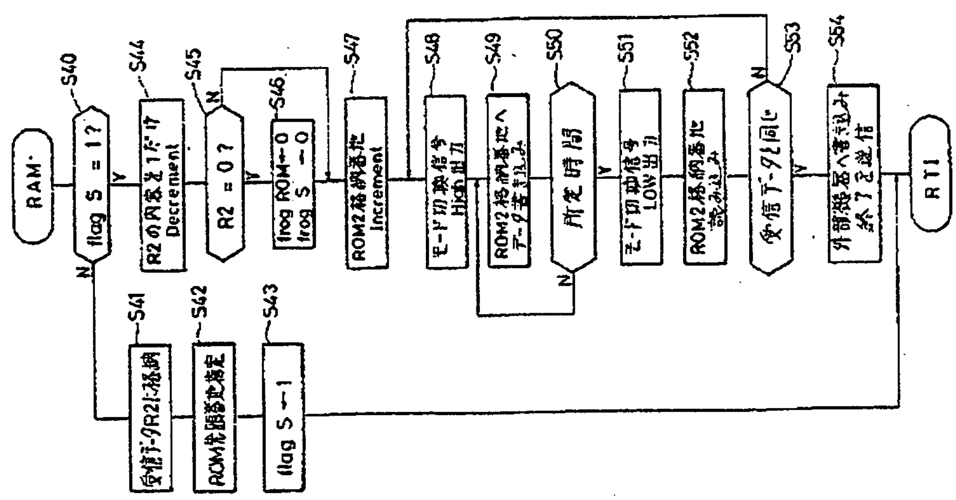


2004/12/10 http://www4.ipdl.jpo.go.jp/tjcontent...gif&N0401...

第 5 図



第 6 図



2004/12/10 http://www.ipdl.ncipi.go.jp/...&N0401

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☑ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: ______________________

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.